

Thoughts on Current and Future Research on Agile and Lean: Ensuring Relevance and Rigor

John F. Tripp
Baylor University
john_tripp@baylor.edu

Jeffrey Saltz
Syracuse University
jsaltz@syr.edu

Dan Turk
Colorado State University
Dan.Turk@colostate.edu

Abstract

Over the past two decades, research in the area of agile and lean software development has mirrored the strong growth of the use of agile and lean methodologies. However, while these research streams have made a significant contribution in the use of agile and lean methodologies, much of the recent research lacks the rigor and relevance to make an impact in research and practice. For example, many of the studies have not measured the actual use of agile or lean methods nor had a significant theoretical grounding. Furthermore, agile research has not expanded to fully cover emerging opportunities and challenges. A deeper theoretical motivation on agile and lean software development can help demonstrate how the principles of, for example, agile software development, may be transferred to these other areas, and hence, broaden the research's relevance. This paper provides commentary intended to help push the agile and lean research agenda forward, and outlines three key criteria that future researchers should consider when conducting research on the phenomenon of agile. The paper also provides an example for the use of the criteria, and presents several initial, open research questions that could help increase the use of agile, including the use of agile and lean concepts in other IT and non-IT contexts.

1. Introduction

It has been more than fifteen years since the use of an agile framework was first suggested [19]. During that time, there has been significant adoption of agile principles and methods in organizations and teams. However, research into the use of agile methods has found that, although teams widely claim to use agile methods, actual use of agile practices is rather low, even in teams led by high-profile agile proponents [11]. Further, while research into the effects of the use of agile methods has found impacts such as a reduction

in bugs [25] and higher job satisfaction on agile teams [36], many studies on agile have not measured the use of the agile actual methods or practices, and have often failed to argue theoretically why agile is theoretically different from other project environments, or identified the core of what organizational, customer, project, and team characteristics are necessary in order to apply agile development principles and practices successfully.

The identification of a *theoretical core* of agile philosophy, methodology, and practice remains elusive, but we believe will be a key component in maintaining and enhancing the relevance and promise of research on agile. In addition, particular organizational, customer, product, project, and team characteristics may encourage or discourage the adoption and utilization of parts or all of agile philosophy and practices. These same characteristics make migration to an agile methodology difficult for some organizations, and have led to mixed success [37]. Characteristics such as strict, engineering based risk management may limit the ability for agile practices and principles to be fully utilized, or require the violation of some agile practices, such as reduced up front planning [16, 22]. Hence, even though agile methods have been widely adopted, there are still many open questions in terms of when and how to leverage agile principles in software development projects as well as in other domains, such as data science efforts.

In this paper, we argue that, for research on agile and lean development to progress, specific attention to rigor must be applied and the relevance and breadth of the research considered. Specifically, in the next section, we describe three key criteria that we believe researchers should consider while performing research on agile. We then provide an illustrative example to assist researchers when using our criteria to approach future research on the agile phenomenon. Finally, to help demonstrate the breadth of open questions in the field, we discuss some research challenges that need to be addressed and how our three criteria can help shape

these research projects – in software development as well in other domains of interest.

2. Rigor in the Study of Agile

Over the years, there have been many claims made as to the benefits of agile versus traditional processes. While anecdotal evidence is regularly presented as to the impacts of agile, there remains very little peer-reviewed, published, empirical evidence supporting these claims and substantiating measureable outcomes of agile vs traditional approaches. This is especially true if we consider that many empirical studies do not measure agile, but rather, simply note that teams claim to be using agile.

We argue that in order for the study of agile development and methods to progress, more empirical studies are needed that possess a set of minimal characteristics. These studies should:

- 1) Describe and measure the team and environmental characteristics of the project,
- 2) Measure the use of multiple agile practices, either qualitatively or quantitatively, and
- 3) Illustrate theoretically how and when the unique nature of agile methods influences outcomes.

We will discuss below why we believe these characteristics are required to be present in order for researchers and practitioners to illustrate whether differences and benefits (e.g., between different agile frameworks, or between an agile framework and a waterfall process) actually exist, and to what extent, and in what circumstances.

While such studies are admittedly difficult to carry out, rigorous, empirical studies are critical to optimizing the use of agile techniques and determining when/where/how such techniques should be adopted within a software development team as well as to understand if such techniques could be used in other situations. This rigor can also increase management's ability to make informed decisions regarding when to use agile versus traditional processes and what outcomes might be expected.

2.1 Characteristics

There are several team and environmental characteristics that drive the extent to which agile methods and practices can achieve their full potential. Hence, when conceptualizing the potential impact of agile methods, researchers must consider and document the characteristics of the environment that enable agile practices to be successfully implemented.

For example, the higher that a problem is characterized by decomposability (the extent that a problem can be reduced to smaller components of

deliverable value), the higher the degree to which agile paradigms such as early and repeated delivery can be realized [3, 2b]. In addition, constraints to the order of production, and requirements that force the delivery of bigger portions of product at a time are key forces that could drive the reduced ability for agile practices to meet their potential. One can note that these examples are applicable in software development as well as in other domains beyond software development. While there are other characteristics that may be salient to the application of agile philosophy and practice, we use these as example cases to help explain our first criterion:

Criterion 1: Directly document and report on the characteristics of the project, team environment, and organizational environment, and how the use of agile was impacted by, and/or impacted those characteristics.

In our opinion, this criterion has been most often met in the research literature on agile. Many examples exist of papers that measure and describe the environment of agile project execution. Numerous case studies exist that have explored in depth the use of agile in environments with high and low environmental complexity (e.g., [25]), teams from a single organization or multiple organizations (e.g. [34]), or teams in a single location vs. teams that are distributed (e.g. [27]).

However, we believe that for research on agile to progress, researchers must engage more deeply with the source theories that they choose to use as a lens through which to view the phenomenon.

2.2 Measurement

Thousands of published papers have studied the impacts of agile development on such dependent variables as coding quality, project success, teaming, and individual satisfaction. However, the literature on agile development has varied widely on how it treats the phenomenon of agile.

Agile has sometimes been conceptualized as a monolithic phenomenon. Because of this, researchers have typically asked teams if they were “using agile”, or have studied teams using a particular method such as XP, or a particular practice such as pair programming (e.g., [30]). In many cases, these papers do not measure the level of use of agile methods or practices, but rather, just the high level concept of use of that method. However, more recent research has shown that the impact of these agile practices (such as using pair programming) may take both direct and indirect paths to a dependent variable (such as code quality) [36]. Furthermore, the measurement of agile practices as opposed to the use of a “methodology” has

become more common [e.g., 25, 36]. This change of focus from method to practice is appropriate; as research shows that few teams adopt agile methodologies as a whole, but rather tailor them to their individual situations [11]. We argue then for our second criterion of rigorous research on agile.

Criterion 2: Directly measure the level use of agile practices.

Multiple recent studies have developed measures for the use of agile practices [e.g., 25, 36], making this criterion of our rigorous research model more accessible for future research. We argue that a rigorous study of agile development requires the measurement of the use of individual agile practices, even in qualitative settings, as the phenomenon that is being studied should be measured.

2.3 Theory

Researchers should examine agile methods from within a theoretical framework or conceptual model that is useful for explanation and prediction [20]. However, much of the literature on agile methods has been weak the use of theory to help explain and generalize research findings. In other words, the research lacks a connection between the agile practices and the theoretical foundations that can help explain those practices [16]. Although multiple theories have been applied in the study of agile, such as the Job Characteristics Model (e.g., [36]), Control Theory (e.g., [25]) and Complex Adaptive Systems (e.g., [26]), no theory has been widely adopted, and these theories have mainly been used to describe what agile is and is not [20]. As such, research into the underpinnings of the core of agile outcomes has lagged [16]. We echo and reassert previous calls for the adoption of rigorous theoretical lenses through which to investigate agile (e.g., [16]).

Further, in order to develop the link between theory and its' underpinning to agile, we argue that it is insufficient to simply utilize existing theory when investigating the agile phenomenon. Researchers instead must argue why agile is different from other contexts and identify the *boundary conditions* that makes agile different. Identifying the boundaries of the applicability of a theory is a key component of the validation of those theories [38]. Boundaries help to identify the internal and external environmental conditions within which the components of a theory or model will hold [20, 38]. Therefore, arguing theoretically when, how, and why agile methods and practices will result in different theoretical outcomes vis a vis other methodologies will help the field identify the core of agile theory. We assert that until

the core theory of agile development is identified, it will be difficult to clearly apply the learning of agile software development to other areas. This concept is stated more formally as criterion 3:

Criterion 3: When applying theoretical lenses, use the nature of agile methods, combined with the nature of the environment to argue for boundary conditions for theory (e.g., what makes agile “different”).

This is the criterion that we believe offers the greatest opportunity for development of theory on agile, and also the criterion that has been most neglected. One key exception is the use of the concept of uncertainty. Uncertainty has been used in multiple studies to establish a boundary condition, specifically that the impacts of the use of agile methods will be higher in environments characterized by higher levels of uncertainty (e.g., [23, 25]). This example of an environmental factor – exogenous to the team is one excellent example of identifying a boundary condition.

We believe that these three criteria should not be used as a checklist, nor do we argue for a type of orthodoxy but, instead, should be used to drive a new focus on agile research. We hope that researchers may explore how these criteria could be used to improve the usefulness and applicability of agile research.

2.4 Example Using the Criteria

In this section, we provide an example of how to utilize the criteria for the development of theoretical boundary conditions for the study of agile methods.

One key theory that has been utilized repeatedly in the research on software development methods is contingency theory [e.g., 7]. Contingency (or fit) theory, when applied to the context of software development, argues that when particular environmental, project, and team contingencies have a “better” fit with the software method in use, it will be associated with “better” outcomes. This is illustrated in figure 1.

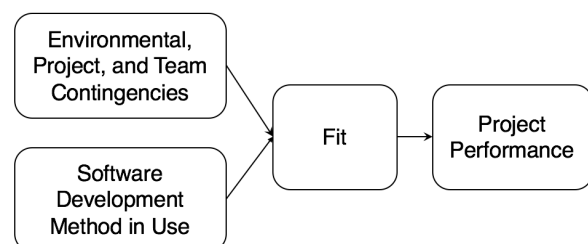


Figure 1. Contingency Theory in Software Development

We argue that one manner through which one could push theorizing on agile forward is to identify, measure, and analyze the effects of the environmental, project, and team contingencies that drive better or worse fit with agile methods. Further, researchers could identify and measure multiple appropriate dependent variable effects that are associated with (or caused by) the use of agile methods. In order to illustrate this, we turn to an old theory of software development, Conway's law [8]¹.

Conway's law states that "Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure." [12], although others have softened this in the manner of "There is a close relationship between the structure of an organization and the artifacts it builds" [14]. Conway's original formulation was limited to the architecture of the system, in that he argued that the interface and architecture of a software product would reflect the social structure of the software team that produced it. This reflection concept spawned the widely used term "mirroring" to describe the outcome.

Despite the relatively narrow focus of the original, the "softened" version of Conway's law [14] has been utilized broadly in the literature to prescribe multiple behaviors including, aligning the organization and architecture, enforce software modularization, collocate developers, develop and maintain effective communication, and maintain flexibility [6]. If Conway's law is a useful for software development, it should hold in all cases, whether using agile development, or non-agile development methods.

Researchers have utilized Conway's law as a lens to evaluate issues relating to the scaling of agile, including communication and architecture requirements when working with distributed teams (e.g., [27]). Not surprisingly, these studies have argued that agile teams must organize according to Conway's principles. However, if agile development is "different", researchers should seek to find boundary conditions where previous "laws" of software development *do not* hold.

This example illustrates one of our key contentions, namely that proving that agile development is constrained by the same forces that constrain all software development is not necessarily a theoretical contribution. Instead of identifying that agile development methods are constrained in the same manner as other methods, researchers should focus on why the unique organizing and execution principles of

agile methods create opportunities where, for example, organizational history and path dependencies do *not* impact the architecture being designed by the team, or why agile communication practices may mitigate the need to collocate developers, or require teams to strictly divide modules of a system by team boundaries, etc.

In summary, Davis [15] argues that interesting results are those that show that structures that are believed to exist, do not exist, or structures that are thought to be generalized, are in fact, specific and local. Hence, in order for rigorous research to move the field forward, we argue that researchers should strive to find interesting results that illustrate which organizational, environmental, project, and team characteristics create an environment in which the use of agile methods leads to divergent outcomes than would appear in the context of the use of traditional methods.

In the next section, we provide guidance for future research via several examples, where we explore how our criteria can help shape research across a broad array of open questions related to the use of agile.

3. New directions for rigorous agile research

In this section, we present a short discussion on several emerging areas of agile research. For each area, we provide a short description of the opportunity / challenge, and then discuss an example of how our criteria could be used to shape future research.

3.1. Lean

Lean thinking focuses on maximizing value and minimizing waste in production processes. The concept emerged in the early years of the Japanese automotive industry. While there is no commonly agreed upon definition of Lean [33], the five original principles of Lean thinking [29] are:

- 1) *Value* – producing value to the customer. If something uses resources but produces no value, it is considered waste and should be eliminated.
- 2) *Value stream* – making sure that the actions that bring a product from initial vision through implementation provide customer value.
- 3) *Flow* – ensuring that the value stream does not have discontinuities so that activities are organized as a continuous 'flow' enabling smooth deliveries.
- 4) *Pull* - producing products (or part of products) only when they are needed ("just-in-time" availability).
- 5) *Perfection* - continuous improvement to achieve zero defects.

The concept of using lean in software development is fairly new. Unfortunately, it is still difficult to

¹ Conway's law is used merely an example. Multiple "old" theories, for instance, from Fred Brooks, or other sources might have been used for our example.

actually implement Lean Thinking as suggested by the pioneers of Lean Software Development [21]. In fact, a review of recent research reveals that although Lean Software Development is a promising approach, the lack of studies and the dominance of some authors make it difficult to draw reliable conclusions [29].

Each of our criteria can easily be applied for lean research. Specifically, research in this area should describe the characteristics of the project, the team and the organizational environment, and how the use of lean impacted those characteristics (criteria 1). In addition, the research should measure the use of lean in reporting the results of case studies (criteria 2), by for example, measuring whether teams that identify as lean teams actually measure a metric for value delivered to the client, and is that a client defined number vs. purely a measure of throughput. Finally, one or more theories should be used to explain why / when lean is more or less suitable than other methodologies, in other words, to explain the boundary conditions (criteria 3). For instance, how does colocation or distribution of the team impact the ability to achieve lean flow [18, 28].

3.2. Kanban

The use of Kanban is growing within software development teams. Kanban has five key elements [5]:

- 1) Visualize the workflow
- 2) Limit work-in-progress (WIP)
- 3) Manage flow
- 4) Make policies explicit
- 5) Implement feedback loops

Kanban was initially used within a lean context (to visualize work and ensure all work has value). However, others view Kanban as supporting agile processes by minimizing work-in-process, which enables the team quickly and easily adapt and define new tasks as needed.

Kanban proponents claim that Kanban offers improved project visibility, software quality, team motivation, communication and collaboration [1]. In addition, a recent survey of Kanban software development practitioners reported that they perceived Kanban as easy to learn and useful in individual and teamwork [2]. The respondents noted several perceived benefits for using Kanban, such as bringing visibility to work, helping to reduce work in progress, improving development flow, increasing team communication and facilitating coordination. Despite the benefits, participants also identified challenges to using Kanban, such as organizational support and culture, difficulties in Kanban implementation, lack of training and misunderstanding of key concepts [2]. In comparing Scrum and Kanban, Lei, Ganjeizadeh, Jayachandran & Ozcan [24] found that for both Scrum

and Kanban projects lead to the development of successful projects. However, they also found that the Kanban method performed better than the Scrum method in terms of managing project schedule.

While many of these findings are promising, the issues previously noted for existing agile research, and the three criteria described to improve the usefulness of future agile research applies to the research relating to Kanban for software development. Much more research is needed to better understand Kanban, and how teams can or should leverage this approach. For example, a key concept in Kanban is to limit work-in-progress (WIP), but defining what the actual limit should be for the work in progress, for a specific column in a Kanban board, is not easily determined by development teams and often not discussed in research results. How teams implement key Kanban concepts, such as WIP limits, needs to be measured to be of use for future Kanban researchers and practitioners.

This rigor will also help research efforts to focus on more foundational open questions, such as what are the pitfalls or key advantages of using Kanban within a software development context as well as within other contexts such as production support or data science teams.

3.3. Hybrid Methodologies

Rather than just selecting one methodology, there are times when teams integrate multiple approaches into one hybrid methodology. In fact, the thinking and theorizing regarding the application and impact of agile have evolved and, in practice, a myriad of custom and hybrid agile methodologies (e.g., agile scrum, XP + Scrum, Kanban, Scrumban), that utilize some or all of the parts of multiple methodologies, have been adopted and put to use [9]. In general, teams combine these concepts to achieve flexibility and lean thinking, or work to scale Agile to make software development processes more efficient.

For example, integrating the traditional waterfall methodology with an agile scrum process, such as a Water-Scrum-Fall approach [35], is a common approach. In fact, it has been reported that hybrid agile/waterfall methodologies are used in the majority of projects [32]. Scrumban (integrating Scrum with Kanban) is a different example of integrating multiple approaches, as is integrating lean process management (including Lean Startup, Customer Development) principles with Scrum practices.

However, it has been observed that it is often the case that software intensive companies select those elements from Agile and Lean that suit them best, creating their own interpretation of Agile/Lean Software Development [29].

Unfortunately, little is known with respect to these hybrid approaches. Since there are so many variations of this hybrid methodology, it is especially important for future research in this area to adhere to our criteria, especially criteria 2 and directly measure not just the level of agile used, but the type of agile used. With these criteria in mind, research questions such as what are the pitfalls of combining Scrum & Kanban? Should these concepts be combined at the project level or a program level? Are there new ways of organizing and advancing agile practices? As one can see, there are many open questions with respect to these combined methodologies. More research could be focused on performing case studies of these hybrid solutions, as well as exploring the theoretical foundations to understand the success drivers for the different hybrid methodologies and what are the boundary conditions that suggest one of these methodologies are more appropriate than other possible approaches.

4. Discussion and Conclusion

In this paper, we described three key criteria that should be considered when conducting agile research. In addition, we explored how, for example, the integration of lean and agile creates many exciting opportunities but also many open research questions and that these research questions need to be explored with rigor, using the three criteria we have outlined.

We believe that agile in software development is an “instance” of agility in projects. Agile can be used in many other project contexts beyond software development, such as for data science [31], healthcare [39], hardware development [13], supply chain strategies [10] or for information security management [17].

Hence, more work is needed to understand and describe what are the generic classes across these various types of projects. In addition, are key insights generated within one context applicable to the other contexts? In general, more work is needed to be able to describe projects in a consistent manner across domains, so that insights generated from one domain can be appropriately leveraged within another domain.

As we move forward to understand how to best leverage agile and lean concepts, there are many open questions that the research community needs to address. Some foundational questions include:

- What guidance can we provide to create and sustain better agile and lean behaviors and more successful outcomes?
- How can we incorporate other functions, such as architecture and production support, into agile and lean frameworks?

- How can organizations and cultures restructure to support these philosophies?
- What are the measurable outcomes of using agile techniques?
- What additional metrics might a team use to measure team performance?
- What are the measurable differences in outcomes when using traditional vs agile techniques?
- What are ways that we can create a repository of knowledge, experiences, cases and empirical data that could be used by research and industry to leverage and expand our understanding of and practical skills in agile techniques?

These questions are relevant not just for software development efforts, but also for teams focusing on other tasks such as production support or data science challenges. Our hope is that as researchers start to address these questions, the three criteria outlined in this paper are incorporated into the research design and output, such that others can more easily leverage and apply the research results.

5. References

- [1] Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J. (2016, January). Transition of software maintenance teams from Scrum to Kanban. In System Sciences (HICSS), 2016 49th Hawaii International Conference on (pp. 5427-5436). IEEE.
- [2] Ahmad, M. O., Markkula, J., & Oivo, M. (2016, May). Insights into the Perceived Benefits of Kanban in Software Companies: Practitioners' Views. In International Conference on Agile Software Development (pp. 156-168). Springer International Publishing.
- [3] Alqudah, Mashal, and Rozilawati Razali. "Key factors for selecting an Agile method: A systematic literature review." *International Journal on Advanced Science, Engineering and Information Technology* 7.2 (2017): 526-537.
- [4] Ambler, Scott W. "Scaling agile: an executive guide." *Agility@ Scale* Whitepaper (2010).
- [5] Anderson, D. J. (2010). *Kanban: successful evolutionary change for your technology business*. Blue Hole Press.
- [6] Bailey, S. E., Godbole, S. S., Knutson, C. D., & Krein, J. L. (2013, October). A Decade of Conway's Law: A Literature Review from 2003-2012. In *Replication in Empirical Software Engineering Research (RESER), 2013 3rd International Workshop on* (pp. 1-14). IEEE.
- [7] Barki, H., Rivard, S., & Talbot, J. (2001). An integrative contingency model of software project risk management. *Journal of Management Information Systems*, 17(4), 37.
- [8] Brooks Jr, F. P. (1995). *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2/E*. Pearson Education India.
- [9] Campanelli, Amadeu Silveira, and Fernando Silva Parreiras. "Agile methods tailoring—A systematic literature review." *Journal of Systems and Software* 110 (2015): 85-100.
- [10] Christopher, M., Harrison, A., & van Hoek, R. (2016). Creating the agile supply chain: issues and challenges. In *Developments in Logistics and Supply Chain Management* (pp. 61-68). Palgrave Macmillan UK.
- [11] Conboy, Kieran, and Brian Fitzgerald. "The views of experts on the current state of agile method tailoring." *Organizational Dynamics of Technology-Based Innovation: Diversifying the Research Agenda* (2007): 217-234.
- [12] Conway, M. E. "How do committees invent." *Datamation* 14.4 (1968): 28-31.
- [13] Cooper, R. G., 16. Agile-Stage-Gate Hybrids, *Research-Technology Management*, 59(1), pp. 21-29. DOI: 10.1080/08956308.2016.1117317
- [14] Coplien, J. O., & Harrison, N. B. (2005). *Organizational Patterns of Agile Software Development*. Pearson Prentice Hall.
- [15] Davis, M. S. (1971). That's interesting! Towards a phenomenology of sociology and a sociology of phenomenology. *Philosophy of the social sciences*, 1(2), 309-344.
- [16] Dingsøyr, T., Nerur, S., Baliyepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development.
- [17] Dorca, V., Munteanu, R., Popescu, S., Chioreanu, A., & Peleskei, C. (2016, May). Agile approach with Kanban in information security risk management. In Automation, Quality and Testing, Robotics (AQTR), 2016 IEEE International Conference on (pp. 1-6). IEEE.
- [18] Fiore, S. M., Salas, E., Cuevas, H. M., & Bowers, C. A. (2003). Distributed coordination space: toward a theory of distributed team process and performance. *Theoretical Issues in Ergonomics Science*, 4(3-4), 340-364.
- [19] Fowler, M & Highsmith, J. (2001), "The agile manifesto." *Software Development* 9(8) 28-35.
- [20] Gregor, Shirley. "The nature of theory in information systems." *MIS quarterly*(2006): 611-642.
- [21] Janes, A. (2015). A guide to lean software development in action. In Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on (pp. 1-2). IEEE.
- [22] Khan, M. R., Fernandez, W. D., & Jiang, J. J. (2016). Is there such a thing as agile IT program management?, eProceedings of the 11th International Research Workshop on Information Technology Project Management (IRWITPM)
- [23] Lee, G., and W. Xia. "Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility." *Mis Quarterly* 34.1 (2010): 87-114.
- [24] Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 59-67.
- [25] Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, 20(3), 377-399.

- [26] Meso, P., & Jain, R. (2006). Agile software development: adaptive systems principles and best practices. *Information systems management*, 23(3), 19-30.
- [27] Moe, N. B., Šmite, D., Šāblis, A., Börjesson, A. L., & Andréasson, P. (2014, September). Networking in a large-scale distributed agile project. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (p. 12). ACM.
- [28] Rizvi, B., Bagheri, E., & Gasevic, D. (2015). A systematic review of distributed Agile software engineering. *Journal of Software: Evolution and Process*, 27(10), 723-762.
- [29] Rodríguez, P., Partanen, J., Kuvaja, P., & Oivo, M. (2014, January). Combining lean thinking and agile methods for software development: A case study of a finnish provider of wireless embedded systems detailed. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on* (pp. 4770-4779). IEEE.
- [30] Salge CA de Lima, Berente, N. (2016) Pair Programming vs. Solo Programming: What Do We Know After 15 Years of Research? 2016 49th Hawaii International Conference on System Sciences (HICSS).
- [31] Saltz J., Shamshurin I., and Crowston K. (2017). Comparing Data Science Project Management Methodologies via a Controlled Experiment. *Hawaii International Conference on System Sciences*.
- [32] Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040-1051.
- [33] Staats, B., Brunner, D., and Upton, D. (2011). Lean principles, learning, and knowledge work: evidence from a software services provider, *Journal of Operations Management*, vol.29, no. 5, pp. 376-390.
- [34] Stapel, K., & Schneider, K. (2014). Managing knowledge on communication and information flow in global software projects. *Expert Systems*, 31(3), 234-252.
- [35] Theocharis, G., Kuhrmann, M., Münch, J., & Diebold, P. (2015, December). Is water-scrum-fall reality? on the use of agile and traditional development practices. In *International Conference on Product-Focused Software Process Improvement* (pp. 149-166). Springer International Publishing.
- [36] Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign. *Journal of the Association for Information Systems*, 17(4), 267.
- [37] Vijayasathya, L. R., & Butler, C. W. (2016). Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?. *IEEE Software*, 33(5), 86-94.
- [38] Weber, Ron. "Evaluating and developing theories in the information systems discipline." *Journal of the Association for Information Systems* 13.1 (2012): 1.
- [39] Williams, S. J. (2017). Analysis of the COPD Pathway: Lean, Agile and Leagility. In *Improving Healthcare Operations* (pp. 79-94). Springer International Publishing.